

# 基于并行处理技术的分布式 3维数据动态调度策略

赵文吉 段福州 莫映

(首都师范大学资源环境与地理信息系统北京市重点实验室, 北京 100037)

**摘要** 为了缓解单机环境下系统资源紧张的问题,在分析单机模式3维数据显示缺点的基础上,首先提出分布环境下的数据处理和调度方案,然后给出了一个通用的逻辑层面的系统模型,最后分析多种数据动态调度策略与优化方案,并给出了客户方和服务方的处理流程。得出并行处理在一定程度上可降低硬件成本,加快运算速度,更好完成3维数据处理任务。

**关键词** 3维数据 分布环境 动态调度

**中图分类号**: P208 TP311.133.1 **文献标识码**: A **文章编号**: 1006-8961(2007)08-1441-05

## The Strategy for Dynamic Attenuation of Distributed 3D Data Based on Parallel Disposal

ZHAO Wen-ji, DUAN Fu-zhou, MO Ying

(The Key Laboratory of Resource Environment and GIS of Beijing, Capital Normal University, Beijing 100037)

**Abstract** In order to alleviate the problem of systematic resource shortage which happened under the unit environment, we firstly put forward the data processing and adjusting scheme under the environment of distributing on the base of analyzing shortcomings of 3D data showing in the mode of single unit. And then we provide the general systematic models of a logic aspect. Finally we analyzed the dynamic dispatcher's tactics of multi-data and optimized scheme, and put forward the treatment procedure that provide both in customer's side and serve side. Our conclusion is that the parallel processing can reduce the hardware cost to a certain extent, can accelerate the speed of operation and is much easier to fulfill the 3D data processing task.

**Keywords** three-dimension data, distributed environment, dynamic attenuation

## 1 引言

在3维地形地物显示方面,以往的很多应用都是建立在单机模式下。鉴于地形及地物数据的海量特性和处理的复杂性,致使在地形渲染的整个过程中,一方面CPU很有可能一直都是满负荷运行,另一方面系统资源的占用也十分可观,由于这会导致系统不能及时响应,帧频率达不到指定要求,从而影

响系统的实时漫游效果。

为了解决此问题,很多学者及研究人员做了大量的研究工作,并提出了很多方法。比如,各种地形的简化算法、分区块思想的引入、3维显示的裁剪算法、遮挡算法等等。其中LOD(level of detail)技术便是一种典型的基于多分辨率表现的地形简化技术。这些方法中,多数已经趋于完善,即使改进算法,效果也不会十分显著<sup>[1-6]</sup>。

但上述解决方案均以单机环境为前提,即在一

**基金项目**: 国家“863”基金项目(2002AA134074); 国家自然科学基金项目(40471090); 北京市自然科学基金项目(KZ200510028017)

**收稿日期**: 2007-03-28; **改回日期**: 2007-04-28

**第一作者简介**: 赵文吉(1967~),男,副教授,系主任。1998年于长春科技大学获博士学位。主要从事空间信息技术在资源环境动态检测中的应用与国土资源信息化研究。E-mail: zhwenji1215@163.com

个或若干个 CPU 以及有限内存、有限系统资源的环境下,研究如何有效利用系统资源,这样尽管计算机硬件水平不断提高,软件算法也在日趋完善,但 3 维地形地物数据的海量性和复杂性决定了单机系统始终存在处理地形地物数据的瓶颈问题。尤其在场景中 3 维实体种类繁多的情况下,各种简化算法,包括 LOD 技术在内,都不能够起到很好的作用,因为此时 CPU 要同时负担计算和显示两个任务,其工作量是相当大的。

为了缓解单机环境下系统资源紧张的情况和在一定程度上降低硬件成本,一种可行的方案是在网络环境下,由一台(或多台)计算机作为前端,负责显示,而将批量数据处理和计算任务分配到网络中的其他计算机上,再通过多台机器协同工作来共同完成 3 维地形地物数据的调度、处理和显示任务。这样不仅可以缓解单机环境下系统资源紧张的问题,同时在维持相同性能水平的前提下,比使用单机更为廉价。

## 2 模型的提出

众所周知,实现分布式数据处理和调度,需要根据实际情况定义合适的系统模型。目前,在分布式计算领域,有很多广泛流行的体系架构,其主要有微软的 DCOM/COM+, SUN 的 J2EE/EJB 和 OMG 的 CORBA 等,这些架构包罗万象,不仅涵盖了分布应用的方方面面,而且提供了多方面的解决方案。3 维 GIS 显示系统有其自身特点,即其性能和需求不同于其他应用,如果直接将现有的体系架构应用于 3 维数据调度,则势必带有盲目性。因此,选择一种技术架构,其能根据应用系统模型的实际情况进行一定的取舍,以满足 3 维数据调度和应用的需求,是十分必要的。针对 3 维地形地物数据处理与显示的基本特点,本文设计了一个通用的逻辑层面的系统模型。

在分布式 3 维数据调度计算与显示系统中,根据功能与职责的不同,存在客户、服务和服务器 3 种角色。

客户(client)是运行于前端的用作显示的计算机终端,其主要功能是将最终的 3 维地形地物数据展示给用户。除此之外,它还担负一定的数据调度逻辑,比如何时决定向网络中的其他机器请求何种服务。当然,也可以将一些计算任务放在客户端

运行,这要视具体情况而定,此处不做过多细致的限定。

服务(service)是一组彼此相互关联的功能(function)集合,而每个功能则代表了对数据的某种处理。客户通过向运行于网络中其他机器上的服务程序发送功能请求来实现一些数据的计算或调度处理。这种服务的物理分布是灵活的,并且是可调整的。一台机器上可能运行有多个服务,不同机器上也可能运行着同一个服务的多个不同副本。如图 1 所示,服务 B 分别运行于网络中的两台机器上,这两台机器上各自还运行着服务 A 和服务 C。

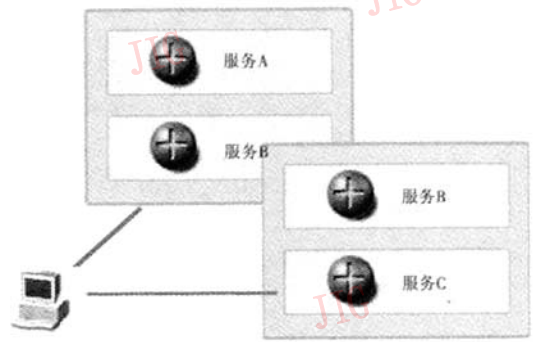


图 1 服务物理分布的灵活性

Fig. 1 Physical distribution services flexibility

服务器管理(service manager)系统要求服务的物理位置对于客户而言是透明(transparent)的。为了达到这个目的,也为了便于管理,需要引入服务管理器的概念。它提供了与服务有关的如下几个功能:

- (1)注册服务(register)
- (2)取消注册服务(unregister)
- (3)申请服务(apply)
- (4)负载均衡(load balancing)

每个服务在部署并启动的时候,需要向服务管理器请求注册,并将服务的自描述信息传送到服务管理器,其内容包括一个标示服务的逻辑名称(可以是一个全局唯一的 ID 号或者一个说明性的字符串)和服务所在的物理位置等。服务管理器用于维护一个服务信息表(service information table),它将在表中插入一条新的记录,把服务的逻辑名称及物理地址绑定(binding)在一起。一方面,当服务停止运行时,它首先需要向服务管理器请求取消注册,然后服务管理器将会把与该项服务对应的记录从表中删

除掉;另一方面,客户在使用服务之前,首先需要向服务管理器发送申请信息,并附上服务的逻辑名称;然后服务管理器根据客户发送过来的逻辑名称,从服务信息表中找到对应的服务所在的物理地址,并返回给客户;最后根据该物理地址,客户才可以向对应的服务发送请求,以便执行具体的数据计算与调度任务。

### 3 数据动态调度策略

基于上述模型,客户所在机器在绘制位于当前视景体内场景的同时,可以根据计算量的大小把计算请求和相应的参数发送给位于后台机器上的服务,使其可以在前端绘制的同时,计算将来可能位于视景体内的场景数据,然后将处理好的数据传回给前端客户,而且客户在绘制完当前场景之后,就可以马上取得下一个场景的已经准备好了的数据。通过这种预先处理的流水线作业模式,可以保证计算与绘制的并行进行。

#### 3.1 基于视点的动态分块策略

为了达到前端绘制和后台计算并行处理的目的,采用了基于视点的动态分块策略,即所谓的分区块思想:首先将整个地形分割为相等大小的  $n \times m$  个数据块(一般是正方形区块),然后每次只显示处于当前视景体内的地形数据块,而在视景体随视点移动而进入下一个数据块的时候,再将对应的数据块调入,并绘制。这种方法很适合于地形数据量很大,以至超过内存容量而无法一次同时处理全部数据的情形。这一策略的另一好处是,将3维地形数据的绘制、加载和计算处理分置于不同的机器上并行处理,由于充分利用了分布环境下的资源与技术,从而达到了提高系统性能的目的。

文献[7]所提供的方法,在开始讨论具体的数据调度策略之前,对照图2,以当前视景体所在的数据块为中心,由内向外将地形区块分成如下不同特征的3种类型:当前可见区、预可见区和将来预可见区。

(1)当前可见区是正在显示的地形区域,即当前处于可见区域(视景体)内的数据块(图2中深灰色区域的方格)。

(2)预可见区是与当前可见区相邻的数据块,一共有8块。当视点移动的位置超出当前可见区的范围,而进入这些相邻区域中的某一个数据块时,则

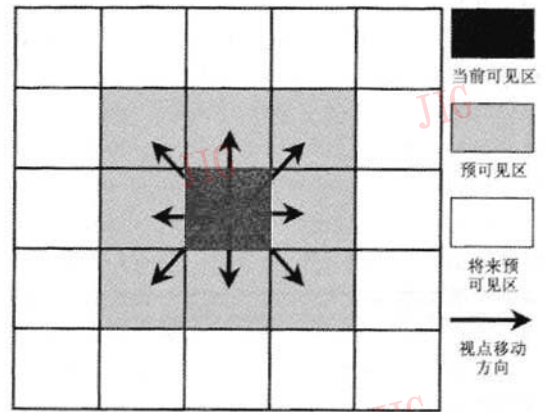


图2 场景数据分块示意图

Fig. 2 Scene data block diagram

由于该区域将成为下一个当前可见区,因此称其为预可见区(图2中浅灰色区域的方格)。

(3)将来预可见区是以当前可见区为准则的下一个可见区(即预可见区中的某一个数据区块),即预可见区中除去当前可见区和对应的预可见区之外的其余数据块,一共有16块(图2中白色区域的方格)。

#### 3.2 场景数据的动态调度策略

当视点移动时,由于与当前可见区相邻的8个预可见区中的一个最有可能成为下一个当前可见区,因此可采用的策略是,在客户方绘制当前可见区的同时,服务方保证对应的8个预可见区的场景数据信息已经加载并处理完毕。这样一来,在客户方显示当前可见区时,将会同时接收到从服务方发来的预可见区数据,并与当前可见区场景的数据一起存放在本地内存当中。而与此同时,服务方在传送数据时又并行地加载和处理好16块将来预可见区的数据。如果由于视点的变化,致使预可见区当中的某一个数据块成为了当前可见区,那么因为此前该数据块已经准备好,并且已位于本地内存当中,所以无需再进行数据获取的工作,而是可以直接从本地内存中取用。与此同时,在显示新的当前可见区的时候,将并行地调入新的预可见区,由于新的8块预可见区数据必定位于原先服务方已经准备好的将来预可见区中,因此服务方只需要根据来自客户方的数据取用请求以及附带的描述视点和场景的参数信息,即可简单地从本地内存中获取对应的数据块,并传回给客户。服务方在传输这些数据块的同时,

将加载和处理好新的将来预可见区的场景数据。

这一过程往复的执行,类似于一种“流水线”的处理方式。通过这种数据调度和处理策略,一方面,可以始终保证客户方在显示当前地形数据的同时,随后就可能将被显示的地形数据直接从本地内存中取用,无需经过网络传输的环节;另一方面,可以始终保证客户方在获取预可见区数据的时候,无需再经过数据的加载和计算(由于此时的数据已经位于服务方的本地内存当中,因而只需要经过必要的网络传输环节即可)。这样整个过程就保证了地形显示的流畅性,引用文献[7]的术语,本文把这种调度算法简称为“1+8+16”策略,即(客户方)显示1块,(客户方)调入8块,(服务方)计算16块。

### 3.3 调度策略的优化

对于“1+8+16”策略,在具体实现时,还可以做进一步的优化处理。例如,以图3为例,图中的当前可见区为R,当视点向右水平移动至某个位置时,预可见区G<sub>3</sub>中的部分内容将会变成可见,当视点继续移动以致G<sub>3</sub>中的可见部分超过了G<sub>3</sub>的一半时,则可以认为,G<sub>3</sub>将由预可见区变为当前可见区,而原来的R则成为了相对于G<sub>3</sub>而言的预可见区。

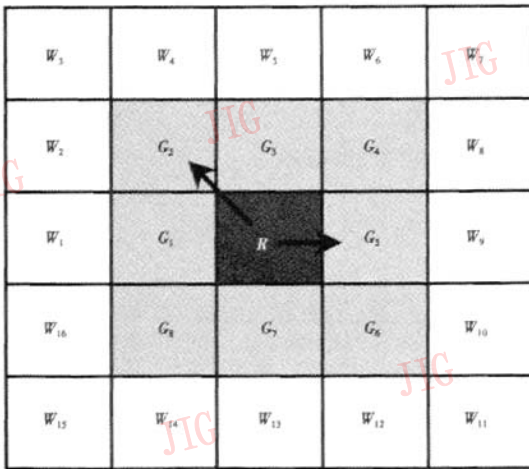


图3 视点沿为水平方向移动得到“1+3+5”策略  
视点沿对角线方向移动得到“1+5+9”策略  
Fig. 3 “1+3+5” strategy in level direction  
“1+5+9” strategy in diagonal direction

此时,对应于G<sub>3</sub>的8个预可见区中,只有W<sub>8</sub>、W<sub>9</sub>、W<sub>10</sub>不在客户方的本地内存中,需要向服务方发送获取请求,而其余的数据块则可以直接从本地内存中获取,并且W<sub>8</sub>、W<sub>9</sub>、W<sub>10</sub>在R作为当前可见区

时,因为已经作为将来预可见区由服务方从外存加载进来,并做过计算处理,所以服务方只需简单地将这3个数据块发送给客户方即可,而对应于G<sub>3</sub>的16个将来预可见区中,只有5块数据区需要从服务方的外存加载进来,并做计算处理,剩下的W<sub>4</sub>~W<sub>7</sub>和W<sub>11</sub>~W<sub>14</sub>总共8块数据区就位于服务方的本地内存中,另外由于还有G<sub>1</sub>、G<sub>2</sub>、G<sub>6</sub>这3个数据块可以直接从客户方的本地内存中找到,因此,在当前可见区从R转入G<sub>3</sub>的时候,只需要采用如下策略即可:(客户方)显示1块,(客户方)调入3块,(服务方)计算5块,将之简称为“1+3+5”策略。

此处是以视点沿水平方向移动为例的,但是对于视点沿垂直方向移动的情况,结论亦类似。而对于视点沿对角线方向移动的情况,若采用类似的分析方法,则将会得到“1+5+9”策略,即(客户方)显示1块,(客户方)调入5块,(服务方)计算9块。假设视点沿图3所示的对角线方向移动,G<sub>2</sub>将成为下一个当前可见区,则除了G<sub>1</sub>、R、G<sub>3</sub>外,客户方还需要向服务方请求获取W<sub>1</sub>~W<sub>5</sub>,以构成G<sub>2</sub>的预可见区,同时除了G<sub>4</sub>~G<sub>6</sub>、W<sub>6</sub>和W<sub>16</sub>外,服务方还需要从外存中加载9个数据块,以构成G<sub>2</sub>的将来预可见区。

### 3.4 客户方和服务方的处理流程

引用文献[7]中的方法来简要描述客户方和服务方的具体处理流程。与文献[7]不同的是,本文的演示程序中,服务程序只是负责将地形数据根据实际需要从外存加载至内存,然后当客户要取用数据的时候再负责将对应的数据块,即当前可见区和预可见区发送给客户,而客户在获得数据之后,再采用基于四叉树的动态LOD算法,对当前可见区中的地形数据进行建模和处理,以达到地形简化的效果。按照“1+8+16”策略,由于此处对将来预可见区的计算处理只涉及数据的加载耗时,因此相对来说是较为简单的。如果数据加载的耗时不是很大的话,则可以进一步将该策略简化为“1+8”策略,即在客户方每次取用预可见区数据的时候,服务方总是从外存调入数据,而非预先准备好。在地形数据3D可视化中这种简单的数据分块处理是最为常见的。

在未来的研究中,还可以给服务增加更多的功能,比如:地形纹理的加载和准备,某些耗时很大的复杂3维运算以及LOD算法本身。

#### 3.4.1 客户方流程

客户方处理流程步骤如下:

(1) 接受来自服务方的首个当前可见区数据,并保存至本地内存;

(2) 接受来自服务方的预可见区数据,并保存至本地内存;

(3) 绘制当前可见区域内的场景;

(4) 搜集交互信息,包括视点位置、视线方向、鼠标位置信息等;

(5) 将上述信息传送给服务方,作为服务方计算下一帧场景绘制数据的依据;

(6) 返回步骤 2。

### 3.4.2 服务方流程

服务方处理流程如下:

(1) 根据系统设置的初始视点位置,加载并处理首个当前可见区以及对应的预可见区;

(2) 将首个当前可见区数据传送给客户方,以供显示;

(3) 将与当前可见区对应的预可见区数据传送给客户方;

(4) 接受来自客户方的数据取用请求,并根据视点位置,视线方向,鼠标位置等信息,计算出与下一帧场景对应的预可见区数据;

(5) 加载并处理对应的将来预可见区数据;

(6) 返回步骤 3。

## 4 结 论

并行处理在一定程度上可降低硬件成本,同时可将批量数据处理和计算任务分配到网络中的其他计算机上,以便通过多台机器协同工作来共同完成

3 维地形地物数据的调度、处理和显示任务。这样做不仅可以缓解单机环境下系统资源紧张的问题,同时,在维持相同性能水平的前提下,比使用单机更为廉价。

### 参考文献 (References)

- 1 Ma Zhi-min, Chen Hao, Wang Jin-ling. Research on the liner octree data structure in three dimensional geographic information systems [J]. Journal of Xian Engineering University, 1999, 21: 55 ~ 60. [马志民,陈浩,王金玲. 三维 GIS 的线性八叉树数据结构研究 [J]. 西安工程学院学报, 1999, 21: 55 ~ 60.]
- 2 Gong Jian-ya, Xia Zong-guo. An integrated data model in three dimensional GIS [J]. Journal of Wuhan Technical University of Surveying and University, 1997, 22(1): 8 ~ 15. [龚健雅,夏宗国. 矢量与栅格集成的三维数据模型 [J]. 武汉测绘科技大学学报, 1997, 22(1): 8 ~ 15.]
- 3 Zhang Hai-yong, Xu You-zhi. 3D model constructing and dynamic display of uncontinuous geological entity based on TIN [J]. Journal of China University of Mining & Technology, 2002, 31(2): 142 ~ 145. [张海荣,许有志. 基于 TIN 不连续地质体的三维构模及动态显示 [J]. 西北工业大学学报, 2002, 31(2): 142 ~ 145.]
- 4 Pouzet J. Estimatin of asurface with known disconten inuities for automaticcontouring purposes [J]. Mathematical Geology, 1980, 12(6): 559 ~ 575.
- 5 Jean Paul L, David M Mark, Lawrence Kikuchi, et al. Two-dimensional run encoding for quadtree representation [J]. Computer, Vision, Graphics and Image Processing, 1985, 30: 56 ~ 69.
- 6 Xu Yuan-bing, Hao Ji-cheng. Algorithm of automatic three dimensional mesh generation by octree data structure technique [J]. Journal of Northeastern University, 1997, 18(4): 351 ~ 355. [徐愿乘,郝冀成. 用八叉树数据结构自动生成三维网格的算法设计 [J]. 东北大学学报, 1997, 18(4): 351 ~ 355.]